Port Scanning

TLDR

```
 # Stealthy
nmap -sS 10.11.1.X

# Scan all ports, might take a while.
nmap 10.11.1.X -p-

# Scan for UDP
nmap 10.11.1.X -sU
unicornscan -mU -v -I 10.11.1.X

# Scan for version, with NSE-scripts and trying to identify OS
nmap 10.11.1.X -sV -sC -O

# All out monsterscan
nmap -vvv -Pn -A -iL listOfIP.txt

# Fast scan
nmap 10.11.1.X -F

# Only scan the 100 most common ports
nmap 10.11.1.X --top-ports 100
```

Nmap

Now that you have gathered some IP addresses from your subdomain scanning it is time to scan those addresses. You just copy-paste those addresses and add them to a file, line by line. Then you can scan all of them with nmap at the same time. Using the `-iL` flag.

Basics - tcp-connect scan

Okay, so a bit of the basics of Nmap and how it works. When one machine initiate a connection with another machine using the **transmission-control protocol (tcp)** it performs what is know as a three-way handshake. That means:

```
 machine1 sends a syn packet to machine2
machine2 send a syn-ack packet to machine1
machine1 sends a ack packet to machine2.
```

If machine2 responds with a syn-ack we know that that port is open. This is basically what nmap does when it scans for a port. If machine1 omits the last ack packet the connection is not made. This can be a way to make less noise.

This is the default mode for nmap. If you do not add any flags and scan a machine this is the type of connection it creates.

"Stealthy" -sS

By adding the `-sS` flag we are telling nmap to not finalize the three way handshake. It will send a `syn`, receive `syn-ack` (if the port is open), and then terminate the connection. This used to be considered stealthy before, since it was often not logged. However it should not be considered stealthy anymore.

In the flag I imagine that the first `s` stands for scan/scantype and the second `S` stands for `syn`.

So `-sS` can be read as **scantype syn**

UDP scan

UDP is after TCP the most common protocol. DNS (53), SNMP (161/162) and DHCP (67/68) are some common ones. Scanning for it is slow and unreliable.

```
 -sU
```

Output scan to a textfile

Not all output works with grepable format. For example NSE does not work with grepable. So you might want to use xml instead.

```
# To text-file
-oN nameOfFile

# To grepable format
-oG nameOfFile

# To xml
-oX nameOfFile
```

Scan an entire IP-range

You might find that a site has several machines on the same ip-range. You can then use nmap to scan the whole range.

The `-sn` flag stops nmap from running port-scans. So it speeds up the process.

```
nmap -vvv -sn 201.210.67.0/24
```

You can also specify a specific range, like this

~~~~~
nmap -sP 201.210.67.0-100
````

#### Sort out the machines that are up

So let's say you find that 40 machine exists in that range. We can use grep to output those IP:s.

First let's find the IPs that were online. Ip-range is the output from previous command. You can of course combine them all.

```bash
cat ip-range.txt | grep -B 1 "Host is up"
~~~~~

Now let's sort out the ips from that file.

```
grep -o '[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}' ip-range.txt > only-ip.txt
```

Now you can input all those ips to nmap and scan them.

Scan a range and output if a specific port is open

Nmap has a command to make the output grepable.

```
nmap -vvv -p 80 201.210.67.0-100 -oG - | grep 80/open
```

Nmap scripts

This chapter could also be placed in Vulnerability-analysis and Exploitation. Because nmap scripting is a really versatile tool that can do many things. Here we will focus on it's ability to retrieve information that can be useful in the process to **find vulnerabilities**

First locate the nmap scripts. Nmap scripts end in `.nse`. For Nmap script engine.

```
locate *.nse
```

The syntax for running a script is:

```
nmap --script scriptname 192.168.1.101
```

To find the "man"-pages, the info about a script we write:

```
nmap -script-help http-vuln-cve2013-0156.nse
```

Run multiple scripts

Can be run by separating the script with a comma

```
nmap --script scriptone.nse,sciprt2.nse,script3.nse 192.168.1.101
```

Run the default scripts

```
nmap -sC example.com
```

Metasploit

We can do port-scanning with metasploit and nmap. And we can even integrate nmap into metasploit. This might be a good way to keep your process neat and organized.

db_nmap

You can run `db_nmap` and all the output will be stored in the metasploit database and available with

```
 hosts
services
```

You can also import nmap scans. But you must first output it in xml-format with the following flag

```
nmap 192.168.1.107 -oX result.xml
```

Good practice would be to output the scan-results in xml, grepable and normal format. You do that with

```
nmap 192.168.1.107 -oA result
```

Then you can load it into the database with the following command.

```
db_import /path/to/file.xml
```

Metasploit PortScan modules

If you for some reason don't have access to nmap you can run metasploits modules that does portscans

```
use auxiliary/scanner/portscan/
```